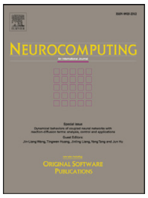
Neurocomputing 384 (2020) 314–324

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



Scale-Recursive Network with point supervision for crowd scene analysis

Zihao Dong[a], Ruixun Zhang[b], Xiuli Shao[a,*], Yumeng Li[a]

[a] *College of Computer Science, Nankai University, Tianjin 300350, China*
[b] *MIT Laboratory for Financial Engineering, Cambridge, MA 02142, USA*

ARTICLE INFO

Article history:
Received 19 April 2019
Revised 13 November 2019
Accepted 16 December 2019
Available online 18 December 2019

Communicated by Mingli Song

Keywords:
Crowd density
Scale-Recursive
Crowd counting
Weakly supervised learning
Joint training

ABSTRACT

Crowd scene analysis, and in particular its density estimation, is a challenging task due to the lack of spatial information, scale variation, and the large amount of supervised-learning parameters. In order to address these challenges, we propose a Scale-Recursive encoder–decoder Network with Point Supervision (SRN+PS). On the one hand, an encoder–decoder recurrent structure uses features between adjacent scales to tackle scale variation, and a novel loss function, called the row vector-based counting loss, is proposed to focus on the crowd counting accuracy. On the other hand, we employ an additional point segmentation task in training and combine features learned from the two tasks above. The Euclidean loss, row vector-based counting loss, and two-label focal loss are integrated by a joint training scheme, which improves both the quality of density map estimation and the performance of crowd counting. Finally, we propose a weakly supervised framework based on the SRN structure and the Convolutional Winner-Take-All(CWTA) module. In this framework, most parameters are obtained by unsupervised learning with the exception of a few which are tuned by supervised learning in model training. As a result, our multi-scale structure can obtain salient object sparse spatial features from unsupervised learning. Experiments on the ShanghaiTech, UCF_CC_50 and UCSD datasets demonstrate the effectiveness of our proposed method.

1. Introduction

Image analysis in crowd scene, including object counting and density map estimation, is an important task in computer vision due to the ever-increasing demand in applications such as traffic congestion detection, product congestion during delivery, and moving objects monitoring. Recently, the literature on crowd scene analysis aims to tackle two main tasks: crowd counting and density estimation. However, high-quality crowd density map (crowd count) prediction is still difficult to achieve due to the complex background and targets with inhomogeneous sizes. Although many state-of-the-art methods [1,2] are used to learn the mapping between pixel level features and density maps (crowd counts), these methods are only based on density supervision and do not fully utilize the semantic segmentation information such as the labeled category of the pixels. A few studies focus on the collaborative learning between the regression model and the segmentation model, and typically thousands of annotations are required because they are based on supervised learning. Therefore, it remains unclear how to efficiently learn the features of unlabeled crowd scene.

Given the above problems, we need to overcome the following challenges: (1) it is difficult to obtain crowd segmentation accurately in most crowd scenes. However, pixel-level features can provide spatial location of predicted crowd objects. Therefore we need to make full use of the segmented feature information to achieve lower estimation errors. (2) The inhomogeneous and extreme-overlapping nature of object instances, high levels of clutter in crowd scene, and scale variation are common problems in images, with the scale variation being the major obstacle. Although a few network architectures [2–4] are proposed for this problem, the features between adjacent scales are not correlated and applied for crowd density estimation. (3) Crowd counting requires large datasets with pixel-level annotation for training (see Fig. 1(a)). In any crowd analysis dataset, each image contains thousands of congestion targets, making the annotation of crowded objects extremely difficult. (4) Recent methods [2,5,6] generate density maps that are inconsistent with the size of the input images. The resizing operation of the density map prediction leads to blurred spatial locations of the detected

[*] Corresponding author.
E-mail addresses: 641069042@qq.com (Z. Dong), 1120170132@mail.nankai.edu.cn (X. Shao).

https://doi.org/10.1016/j.neucom.2019.12.070
0925-2312/© 2019 Elsevier B.V. All rights reserved.

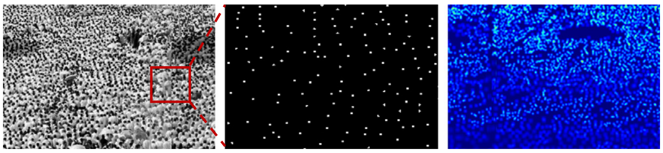

(a) Input Image (b) Point Map Mask (c) Density Map

Fig. 1. Typical dense crowd images with human annotations (from UCF_CC_50 dataset).

objects, leading to higher count error rate and lower quality density maps.

To tackle these challenges, we propose a collaborative learning framework to leverage the diversity of features and unlabeled data for crowd counting. An encoder–decoder recurrent structure, which includes one encoder and two decoders, is employed to extract high-scale density information and low-scale features to address the scale variations. The structure takes a sequence of images with crowd scenes as the input of the encoder at different scales. One of the decoders up-samples the features learned by the encoder to complete the density estimation, and the other decoder segments the foreground points in feature map by point supervised learning. The point supervised module is similar to the attention guided detection module [7], which are both used to detect the location of each person. The above objectives are integrated via a collaborative learning strategy. In order to address the third challenge above, we add three convolutional winner-take-all modules [8] between the down-sampling and up-sampling paths, inspired by the GWTA method [9]. This helps convergence in model training with unlabeled data.

Our multi-scale collaborative learning architecture has following advantages compared to the existing literature:

1. Our CNN model is trained by collaborative learning with two different computer vision tasks, namely the density estimation and the point segmentation, as shown in Fig. 1. These two related tasks effectively assist and regulate each other to obtain high-quality density map estimations.
2. A novel l_2 loss function that counts the row vector sums of the density map is proposed to generate more accurate counting results. The fine-tuned model based on a weighted sum of multiple loss functions can overcome the gap between accurate crowd counting and high-quality density map estimations.
3. The CWTA based module used in our encoder–decoder structure provides more information for unlabeled data training. This contributes to the growing literature on weakly supervised learning of crowd scene analysis.

2. Related work

CNN-based methods have achieved great success in numerous computer vision tasks such as semantic segmentation and object detection, which inspired many researchers to train models for corresponding density maps and crowd count. Sindagi and Patel [10] summarize CNN-based methods into four categories based on their network properties. Among them, the multi-task learning framework [11,12] has been most widely used in crowd analysis task recently. The recent literature on crowd scene analysis can be classified into two categories: detection-based (or segmentation-based) approaches and density estimation-based approaches. The former either adopts a moving-window architecture to detect crowded people or uses the processed semantic labels to segment the density regions, and the latter estimates density maps that represent the crowd counts by the sum of pixel values.

Detection-based (or segmentation-based) approaches. In addition to crowd counting in crowd scene analysis tasks, Kang et al. [12] use classification and regression methods to generate full-resolution density maps on detection and tracking tasks. A sliding window regressor predicts the density for every pixel, which improves detection and tracking performances. In another study [13], all congested object instances are viewed as a set of sequences, and an LSTM controller decodes the GoogleNet-encoded features into a set of detections. However, they are only used to count a class of objects. The GMN [14] formulates counting as a matching problem, and labeled video data is used to train for tracking, which can classify multiple instances of objects in an image. In addition, in order to obtain the spatial features of congested objects, Kang and Wang [15] utilize a fast Fully Convolutional Neural Network (FCNN) for crowd segmentation. Although the FCNN only predicts the rough area of congestion, it provides new ideas for crowd scene analysis with additional features such as context and category except crowd counting. Instance regions [16] are used to locate the object in order to count accurately, and this method is similar to image semantic segmentation, which uses points to represent different object instances and divides them into different sized regions. However, it does not perform well in object counting with high density and mutual occlusion. Furthermore, recent studies [17,18] try to learn more useful CNN features that are rotation invariant, which is an important component in object detection methods. We will also consider rotation invariance in our future study of crowd analysis.

Density estimation-based approaches. CNN-based crowd counting and density estimation methods are widely used in crowd scene analysis. Onoro-Rubio and López-Sastre [1] propose a scale-aware counting model, the Hydra CNN, for object density estimation. Similar ideas are applied to a multi-column-based architecture (MCNN) [2] which further improves the prediction performance. In addition to the research on network structure improvements, Huang et al. [19] take advantage of the effective variants of pooling modules, which are called multi-kernel pooling and stacked pooling, to gain high scale invariance. Sam et al. [20] train a CNN-based model with a combination of regression neural networks, and a switch classifier is used to select the best CNN regressor. These models belong to the same category of scale-aware models that are robust to variations in scale. Another set of methods [6,21] integrate contextual information into the CNN framework, which can deal with the problem of complex backgrounds by learning local counts. Recently, inspired by the multi-task learning for crowd counting problems [22], various methods have expanded from two related learning objectives, the crowd density and crowd count, to other learning tasks such as attention maps [23], crowd velocity maps [24], and density level classification tasks [6]. Finally, performance can also be improved by utilizing new loss functions such as the adversarial loss [4] and the perceptual loss [4].

Weakly supervised learning. Most crowd counting tasks require labeled data, but little work has been done in weakly supervised learning for crowd counting. There is a growing body of research in this direction. A standard autoencoder [25] structure only consists of encoder and decoder, but the coding distribution of the encoder output is not processed by other modules. To tack this problem, Makhzani et al. [26] impose a prior distribution on the latent representation to reduce the reconstruction error. However, the architecture that combines GAN and autoencoder is complex, which may not be ideal for data with unbalanced positive and negative samples. The convolutional WTA autoencoders [8] use spatial and lifetime sparsity constraints to optimize the data distribution of the encoder, which can finally obtain more accurate data features.

3. The proposed method

The overall framework of the proposed model, which we call the Scale-Recursive encoder–decoder Network with Point

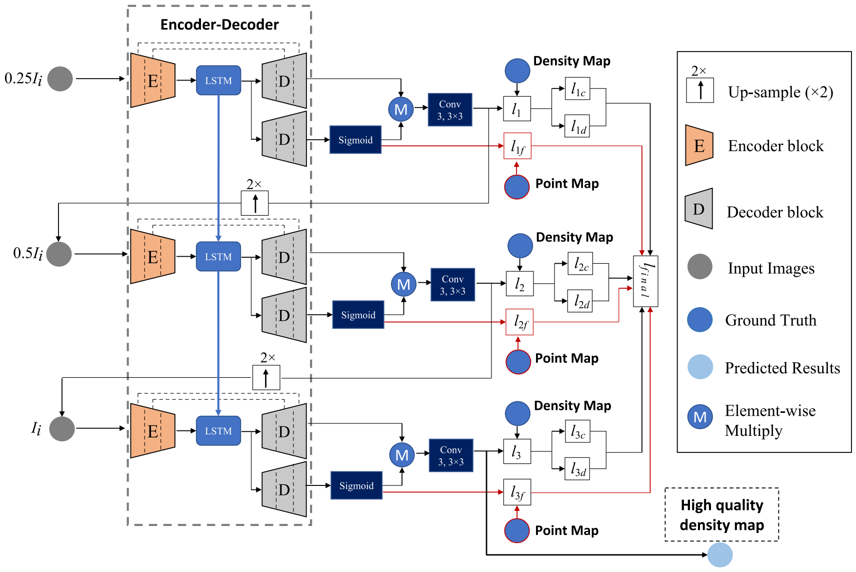


Fig. 2. The architecture of Scale-Recursive encoder–decoder Network with Point Supervision. The red arrow is defined as the training path of the point feature map, and the black arrow is defined as the training path of the density map. The weighted calculation process of each loss function is shown on the right-hand side.

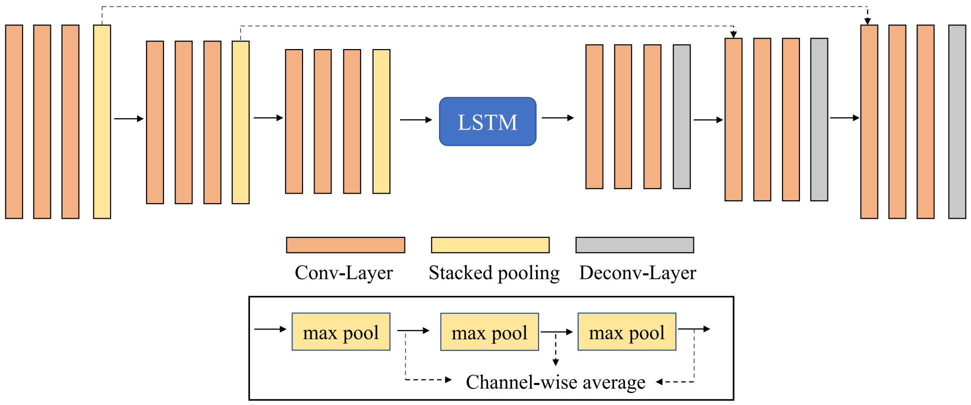


Fig. 3. The framework of proposed encoder–decoder network.

Supervision (SRN+PS), is illustrated in Fig. 2. In the dashed box, the Scale-Recursive structure is adopted across three different scales in the coarse-to-fine strategy, which is only used for performance optimization; each encoder/decoder module (in Fig. 3) takes a sequence of images generated by the previous module as inputs, and a set of corresponding density maps and point maps are produced at different scales. Then element-wise multiplication is applied on the density map and point feature map to generate a refined density feature map. Finally, we weight the loss functions at different scales, and the last scale encoder/decoder module will generate the high-quality density map.

3.1. Encoder–decoder network

In Fig. 3, we use three groups of convolutional layers in the encoder and decoder, respectively. Between the encoder and decoder module, LSTM is added to capture useful information and complete cross-scale recovery. Due to the visual similarity of the regions of different scales in the crowd analysis context, we need effective variants of the pooling module. Instead of the single max pooling, we use the stacked pooling with a set of kernels {2, 2, 3} [19] to improve the scale invariance of convolutional layers, which contains pooling kernels with different receptive fields to capture the features at multi-scale convolutional layers. Therefore its feature maps are computed as

$$Stack(s) = \frac{1}{3}\sum_{i=1}^{3} Pool(k_i, s_i) \tag{1}$$

where $Stack(s)$ is the stack pooling layer with stride s and $s = 2$. $Pool(k_i,\ s_i)$ is the max pooling layer with kernels k_i and strides s_i. Here $k_1 = k_2 = 2, k_3 = 3$ and $s_1 = 2, s_2 = s_3 = 1$. The bottom of Fig. 3 shows the specific structure of stack pooling.

We use $Conv(o,\ k)$ to denote the traditional convolution layer with o outputs and kernel size k, and $Deconv(o,\ k,\ s)$ to denote the de-convolution layer with o outputs, kernel size k, and stride s. The parameters can be represented as: $3 \times Conv(32, 5) - Stack(2) - 3 \times Conv(64, 5) - Stack(2) - 3 \times Conv(128, 5) - Stack(2) - LSTM - 3 \times Conv(128, 5) - Deconv(128, 4, 2) - 3 \times Conv(64, 5) - Deconv(64, 4, 2) - 3 \times Conv(32, 5) - Deconv(32, 4, 2) - Conv(3, 3)$. We also use skip-connections between the corresponding encoder and decoder to combine different levels of information (see dash parts in Fig. 3).

3.2. Scale-Recursive Network with point supervision

In contrast to MCNN [2], we adopt a new recursive structure across three scales in coarse-to-fine strategy. We form the generation of a density map at each scale as a sub-problem of our crowd

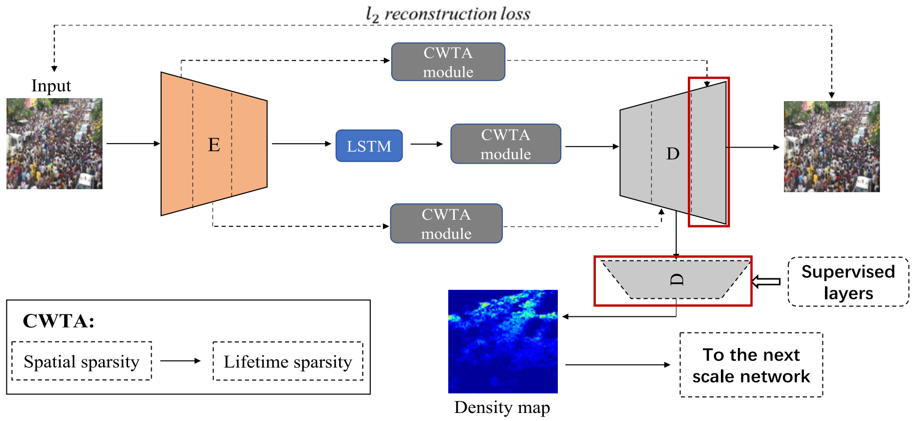


Fig. 4. The single scale architecture of the proposed CNN based on weakly supervised learning.

scene analysis task, which takes an image with crowd information and a predicted result (upsampled from previous scale) as input, and estimate the density map at this scale as

$$\mathbf{D}^{i+1}, \mathbf{H}^{i+1} = N_{SRN}(\mathbf{I}^{i+1}, \mathbf{D}^{i}, \mathbf{H}^{i}; \theta_{SRN}) \tag{2}$$

where i is the scale index, with $i = 3$ representing the final and finest scale. $\mathbf{I}^i$ and $\mathbf{D}^i$ are the input images and estimated density maps at the ith scale respectively. N_{SRN} is the proposed Scale-Recursive network with training parameters expressed as θ_{SRN}. The hidden state H_{i+1} can be transmitted between adjacent scales, and be used to obtain the initial density map and features from previous coarse scale (ith scale). These estimated maps in different scales follow a recurrent relationship, which is sharing network weights to reduce the number of trainable parameters and training difficulty. In addition, they can capture and fuse useful feature information across scales.

Eq. (2) gives the general definition of the network. We describe the details here because several changes are needed to apply this encoder–decoder networks to our framework. Firstly, we insert the LSTM module between the encoder and decoder to improve the overall performance. Because ConvLSTM [27] performs better in our experiments, it is chosen as the core module of this part. Secondly, our Scale-Recursive network requires point map mask (see Fig. 1(b)) supervision to obtain the crowd segmentation information. Similar to the architecture of density regression, we use the same encoder and another decoder with the same structure. In the labels of the point map, the value of each background pixel is set to 0, while the value of each object class is set to 1. The modified network with Point Supervision is expressed as

$$\begin{aligned}
&\mathbf{F}^{i+1} = N_E(\mathbf{I}^{i+1}, \mathbf{D}^{i}; \theta_E) \\
&\mathbf{H}^{i+1}, \mathbf{L}^{i+1} = ConvLSTM(\mathbf{H}^{i}, \mathbf{F}^{i+1}; \theta_{LSTM}) \\
&\mathbf{D}_1^{i+1} = N_{1D}(\mathbf{L}^{i+1}, \mathbf{G}_D^{i+1}; \theta_{1D}) \\
&\mathbf{S}^{i+1} = N_{2D}(\mathbf{L}^{i+1}, \mathbf{G}_S^{i+1}; \theta_{2D}) \\
&\mathbf{D}^{i+1} = Conv(\mathbf{D}_1^{i+1} \times \mathbf{S}^{i+1})
\end{aligned} \tag{3}$$

where N_E is the encoder CNNs with parameters θ_E, N_{1D} is the decoder CNNs of density map estimation path with parameters θ_{1D}, and N_{2D} is the decoder CNNs of point map prediction path with parameters θ_{2D}. The final density map $\mathbf{D}^{i+1}$ is the convolution of the point map $\mathbf{S}^{i+1}$ multiplied by the density map $\mathbf{D}_1^{i+1}$. $\mathbf{G}_D$ is defined as the ground truth of density map, and $\mathbf{G}_S$ is defined as the ground truth of point map. Three encoders and decoders at different scales are used in N_E and N_D, respectively. The hidden feature $\mathbf{H}^i$ contains useful density information after the *ConvLSTM* processing, which is passed to the next scale $\mathbf{H}^{i+1}$. In our experiment, the ith scale is set to half the size of the $(i+1)$th scale.

3.3. Scale-Recursive Network based on weakly supervised learning

To explore the application of weakly supervised learning in crowd scene analysis, we still use our proposed Scale-Recursive Network and train almost all parameters with unlabeled data, followed by supervised training of the retaining parameters. The remaining layers are trained with an unsupervised method called the Convolutional Winner-Take-All (CWTA) [8]. Because most parameters are obtained by the unsupervised learning approach, we call it the Scale-Recursive Network based on weakly supervised learning.

Fig. 4 shows the proposed single scale architecture for our CNN based on weakly supervised learning, and the other two scales are based on Fig. 4 and connected together according to Fig. 2. Compared with the encoder–decoder structure in Fig. 3, the CWTA module is added between three up-sampling and down-sampling paths respectively. This helps our model with more efficient training, and achieves feature sparsity including both spatial and lifetime sparsity, which is better for highly diverse crowd data. The encoder–decoder path of Fig. 3 is used for unsupervised training, and only the last 3 convolutional layers of the decoder on the last scale are trained with less labeled data to tune the parameters and output density feature map.

CWTA Module. Following the method in Makhzani and Frey [8], we replace its auto-encoder with our encoder–decoder network with skip-connection. The CWTA module is trained under two winner-take-all sparsity constraints: spatial sparsity and lifetime sparsity. To regularize the encoder–decoder network effectively, spatial sparsity only chooses the largest hidden activity within each feature map to generates a sparse representation. However, it is time-consuming in the reconstruction of each image. In order to further increase the sparsity, the winner-take-all lifetime sparsity is exploited to avoid the dead filter problem that often occurs in sparse coding, which forces every filter to be updated upon visiting every mini-batch. Through these two constraints, the CWTA module can be used for learning hierarchical sparse representations in this weakly supervised learning task.

3.4. Loss function

Full supervision. In the density map estimation path of Fig. 2, we first adopt the Euclidean loss l_d for each scale to measure the distance between the predicted density map and the ground truth, which is the sum of l_{id} loss at each scale,

$$l_d = \sum_{i=1}^{n} l_{id} = \sum_{i=1}^{n} \frac{\lambda_{id}}{M_i} ||D^i - G_D^i||_2^2 \tag{4}$$

where D^i and G_D^i are the prediction of our network and ground truth of the density map respectively in the ith scale. λ_{id} is the weight for each scale, which we set to 1 after several experimentation. M_i is the total number of pixels in D^i, and n is defined as the number of scales. In Fig. 2, our SRN contains 3 scales, so n is set to 3.

In addition to the density map regression by L_2-norm, we notice that regression of the crowd count greatly improves the performance of estimated performance, which is likely because the congestion levels of images are different. For example, in the sparse scenario, the head count is usually not very large. Row vector-based counting loss can solve the imbalance of sparse and dense degree in crowd scene, where the density map D^i is defined as a set of row vectors: $D^i = [\mathbf{D}_{r1}, \mathbf{D}_{r2}, \ldots, \mathbf{D}_{rn}]^T$. Therefore, this loss function can be expressed as

$$l_c = \sum_{i=1}^{n} l_{ic} = \sum_{i=1}^{n} \frac{\lambda_{ic}}{M_i} || \frac{sum(D^i, 1) - sum(G_D^i, 1)}{sum(G_D^i, 1) + 1} ||_2^2 \tag{5}$$

where $sum(D^i,\ 1)$ and $sum(G_D^i, 1)$ are the estimated and ground truth human count by summing the density map by row. The weight λ_{ic} is set to 1 in the same way as Eq. (4). One is added to the denominator to avoid division by zero.

In addition to the above loss function of the density map path in our proposed network, we introduce another point map loss function in the point map path training process. The point map loss function is a two-label focal loss [28], defined as

$$\begin{aligned} l_f &= \sum_{i=1}^{n} l_{if} = \sum_{i=1}^{n} \frac{\lambda_{if}}{M_i} FL_i \\ FL_i &= -\alpha(G_s^i - P^i)^{\gamma} \log(P^i) - (1-\alpha)(P^i)^{\gamma} \log(1 - P^i) \\ P^i &= sigmoid(S^i) \end{aligned} \tag{6}$$

where G_s is the point map ground truth, and P_i is the probability of each pixel in predicted point map activated by sigmoid function. As suggested by previous research [28], we set $\alpha = 0.25$ and $\gamma = 2$ to balance the distribution of positive and negative categories respectively.

Following the method in Shen et al. [4], we also add the perceptive loss function [29] l_p to minimize the perceptual differences between the predicted density map and ground truth. The Scale-Recursive Network with point supervision is trained using the following final loss function

$$l_{final} = l_d + \beta_1 l_c + \beta_2 l_f + \beta_3 l_p \tag{7}$$

where β_1, β_2, and β_3 are weighting parameters that are all set to 0.1 in the experiments. We also include an experiment in Section 4.6 to analyze the sensitivity of these three parameters. We use this multi-task combined loss function to do joint training in fully supervised learning.

Weak supervision. In the unsupervised training part, Q_w^i denotes the output of the encoder–decoder network (see Fig. 4) at the ith scale, and $Q_w^{'i+1}$ is the corresponding decoder reconstruction at the $(i+1)$th scale. Therefore the loss function is given by

$$l_w = \sum_{i=1}^{n} \frac{1}{M_i} ||Q_w^i - Q_w^{'i+1}||_2^2. \tag{8}$$

In the supervised training stage, the supervised layers (see the dotted box at the bottom of Fig. 4) are trained to minimize the regression loss between the predicted and ground truth density map. Here the regression loss function is defined as

$$l_w^D = \sum_{i=1}^{n} \frac{1}{M_i} ||D_w^i - G_w^i||_2^2 \tag{9}$$

where D_w^i and G_w^i stand for the estimated density map and the corresponding ground truth density map at the ith scale.

4. Experiments

In this section, we perform experimental verification and compare results on three challenging public datasets: ShanghaiTech [2], UCF_CC_50 [30] and UCSD [31]. In training, we set the learning rate to be $1e-5$, bach size to be 4, and the training epoches to be 1400, 200, and 20 on the three datasets, respectively. We evaluate the performance by two common metrics: mean absolute error (MAE) and mean square error (MSE), which are defined as:

$$\begin{aligned} MAE &= \frac{1}{N} \sum_{i=1}^{N} |sum(D_i) - sum(G_i)| \\ MSE &= \sqrt{\frac{1}{N} \sum_{i=1}^{N} |sum(D_i) - sum(G_i)|^2} \end{aligned} \tag{10}$$

where N is the number of images in test set, D_i and G_i are the predicted and ground-truth density map respectively, and the sum of these two matrices is the total count of prediction and ground truth.

Following Sindagi and Patel [21], we also use two standard metrics, PSNR (Peak Signal-to-Noise Ratio) and SSIM (Structural Similarity in Image), to evaluate the quality of the density map of each image. They are defined as:

$$\begin{aligned} &PSNR = 10log_{10}\left(\frac{(2^n-1)^2}{MSE}\right) \\ &SSIM(D_i, G_i) = l(D_i, G_i) \times c(D_i, G_i) \times s(D_i, G_i) \end{aligned} \tag{11}$$

where n is the number of bits per pixel, and $l(.)$, $c(.)$, and $s(.)$ are image similarity measures from brightness, contrast, and structure respectively.

4.1. Density map for training

Both training and testing require crowd images and their corresponding ground-truth density maps. However, almost all crowd datasets only contain the coordinates of points that represent humans, so the conversion from point sets to density maps is required. Following the same scheme in Shen et al. [4], we use the distance matrix to determine the head radiuses. To deal with head size variations and perspective distortions, we utilize the adaptive Gaussian kernels instead of the traditional Gaussian kernels to generate robust density maps. Fig. 5 shows two density maps by traditional Gaussian kernels and adaptive Gaussian kernels generated from the ShanghaiTech Part A dataset. The color bar at the top of the figure represents the distribution of values in the density maps, with values decreasing from left to right. The red boxes in each plot show the quality of density maps generated by the two different methods. In density crowd scenes, crowd density maps obtained by convolving adaptive Gaussian kernels have higher resolution and better distinguishes different crowd heads.

The variance σ_i is used to describe the shape of the distribution, which is related to the distance between the object and k neighbors generated by the Kd-Tree. Larger values of σ_i imply greater fluctuations in the data, and these nodes do not belong to the same category space, so the Kd-Tree needs to be partitioned in this node. Here we only calculate the average distance d_i between four neighbor nodes and the target object, and the variance σ_i is defined as $0.3d_i$. The generated density map is a convolutional value, which is calculated as

$$G(x) = \sum_{i=1}^{N} \delta(x - x_i) * GA_{\sigma_i}(x) \tag{12}$$

where x_i is the object location, $\delta(x - x_i)$ is equivalent to an impulse

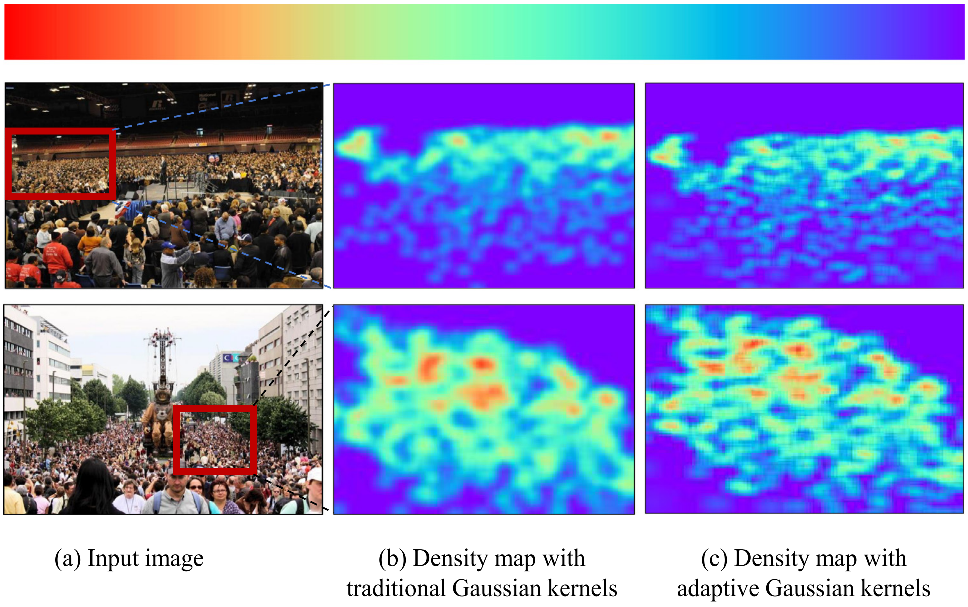

(a) Input image
(b) Density map with traditional Gaussian kernels
(c) Density map with adaptive Gaussian kernels

Fig. 5. Representative density maps from the ShanghaiTech Part_A dataset.

function, and $GA(.)$ is the Gaussian function. However, the adaptive Gaussian kernel method is only applicable to dense crowd scene analysis. In contrast, the fixed Gaussian kernel approach is applied to the opposites of density estimations due to the wide distance between objects.

4.2. Datasets

ShanghaiTech [2] is divided into two parts: Part_A and Part_B. It contains a total of 1198 labeled images in different crowd scenes, and the crowd images in Part_B are sparser than those in Part_A. We use 300 images for training and 182 images for testing for Part_A; 400 images for training and 316 images for testing for Part_B. Because our encoder–decoder structure requires the input images and output density maps to have a fixed size, we resize all images and ground truth including density maps and point maps to 720 × 720. To augment the training data, 240 × 240 patches are randomly cropped and folded from different locations. In the testing stage, we merge some 240 × 240 patches into 720 × 720 images that represent the predicted density map.

UCF_CC_50 [30] is a typical dense crowd counting dataset, which only has 50 annotated crowd images with different levels of congestion. Following Idrees et al. [30], we use 5-fold cross-validation. For each validation, forty images are used as training samples and the remaining ten as the validation set. To augment the training data, each image is randomly cropped to nine 176 × 176 patches, which generates a total of 360 patches in one fold of the training data. In the testing stage, to evaluate each image fairly, the nine cropped patches are combined to calculate the final evaluation metrics.

UCSD [31] contains 2000 labeled frames of size 158 × 238. Compared to UCF_CC_50, it is a sparse crowd counting dataset with the largest count of 46. The provided ROI for each video frame helps reduce distractions from the complex background, and the pixels outside of ROIs are set to zero. Following past literature [2,31], frames from 601 to 1400 are used as training data, and the other 1200 frames are used as testing data. We set the variance σ_i of the gaussian kernel to a fixed value 4.0, and the same data augmentation approach as the ShanghaiTech dataset is adopted to prevent overfitting.

Table 1
Estimation results on the ShanghaiTech dataset.

Dataset	Part_A		Part_B	
Methods	MAE	MSE	MAE	MSE
Cross-scene [22]	181.8	277.7	32.0	49.8
MCNN [2]	110.2	173.2	26.4	41.3
Cascades-MTL [32]	101.3	152.4	20.0	31.1
Switching-CNN [20]	90.4	135.0	21.6	33.4
ACSCP [4]	75.7	**102.7**	17.2	27.4
SRN+PS (ours)	75.0	115.2	**13.8**	**18.8**
CP-CNN [21]	**73.6**	106.4	20.1	30.1

Table 2
Estimation results of crowd count on the UCF_CC_50 dataset.

Methods	MAE	MSE
Cross-scene [22]	467.0	498.5
MCNN [2]	377.6	509.1
Hydra-2s [1]	333.7	425.7
Hydra-3s [1]	465.7	371.8
Cascades-MTL [32]	322.8	397.9
Switching-CNN [20]	318.1	439.2
CP-CNN [21]	295.8	**320.9**
ACSCP [4]	291.0	404.6
SRN+PS (ours)	**289.7**	384.2

4.3. Comparisons with state-of-the-art methods

In order to evaluate the performance of our approach, we compare it with the previous state-of-the-art methods in three different public datasets. The results of our method trained in fully supervised fashion are shown in Tables 1–3.

ShanghaiTech-A and UCF_CC_50 are dense crowd datasets. As shown in Tables 1 and 2, SRN+PS performs better than most of the state-of-the-art methods on ShanghaiTech-A. The cross-scene method [22] gives the direction of density estimation: estimating the counts of crowd people by density map. Based on Zhang et al. [22], the MCNN method [2] achieves robust and better performance through multi-column structures. The Cascades-MTL [32] jointly learns the crowd count classification and the density map, achieving performance improvements without using multi-column networks. However, the category information

Table 3
Estimation results of crowd count on the UCSD dataset.

Methods	MAE	MSE
FCN-MT [33]	1.67	3.41
Cross-scene [22]	1.60	3.31
Switching-CNN [20]	1.62	2.10
FCN-rLSTM [34]	1.54	3.02
CCNN [1]	1.51	–
SRN+PS (ours)	1.24	1.63
MCNN [2]	**1.07**	**1.35**

Table 4
Estimation results of crowd count based on weakly supervised learning on the ShanghaiTech Part_A dataset.

Dataset	Methods	MAE	MSE
Part_A	CCNN supervised [9]	124.6	186.9
	Autoencoder [9]	162.1	233.3
	GWTA-CCNN [9]	**154.7**	229.4
	SRN+CWTA(Ours)	158.7	**223.3**
UCF_CC_50	CCNN supervised [9]	367.2	551.3
	Autoencoder [9]	1272.8	1166.2
	GWTA-CCNN [9]	433.7	583.3
	SRN+CWTA (ours)	**364.2**	**459.1**

Table 5
The quality of density maps on the ShanghaiTech Part_A dataset.

Methods	PSNR	SSIM
MCNN [2]	20.91	0.52
CP-CNN [21]	21.72	0.72
SRN+PS (ours)	**22.74**	**0.78**

Table 6
The quality of density maps generated by SRN+PS in all three datasets.

Dataset	PSNR	SSIM
ShanghaiTech Part_A [2]	22.24	0.78
ShanghaiTech Part_B [2]	24.17	0.83
UCF_CC_50 [30]	13.61	0.25
USCD [31]	17.82	0.80

Table 7
A comparison of the numbers of parameters (in millions) for each method.

Methods	Cross-scene [22]	MCNN [2]	Switching-CNN [20]
Parameters	22.5	0.13	15.1
Methods	CP-CNN [21]	ACSCP [4]	**SRN+PS**
Parameters	68.4	5.1	**13.2**

Table 8
The ablation experiment of our method on the ShanghaiTech Part_A dataset.

Methods	MAE	MSE
SRN(l_d)	129.6	179.5
SRN+PS(l_d)	100.2	154.3
SRN+PS(l_d, l_c)	83.3	136.4
SRN+PS(l_d, l_c, l_p)	75.0	115.2
SRN with single pooling	134.2	194.1
SRN with stack pooling	129.6	179.5

can only provide image-level information in training, and our method SRN+PS makes full use of the pixel-level segmentation information. As a result, our method achieves 26% lower MAE than Cascades-MTL on ShanghaiTech-A and 10.3% lower MAE on UCF_CC_50 dataset. The Switching-CNN [20] uses a switch classifier to choose CNN regressors to get better performance than the MCNN [2]. Apart from scale-aware methods, the adversarial loss is adopted to improve the quality of estimated density maps in Shen et al. [4]. However, generators on a small scale cause feature loss, and our method cascades encoder–decoder networks at three different scales to tackle this problem. Although the performance of the SRN+PS is worse (MSE is 12.5 higher) than the ACSCP [4] on ShanghaiTech-A dataset, SRN+PS achieves improvements on more dense crowd examples (UCF_CC_50 dataset) with 1.3/20.4 lower MAE/MSE. The CP-CNN [21] is a better crowd scene analysis framework than previous attempts [32]. It uses contextual and density class information based on the MCNN [2] and performs the best compared to the other methods on ShanghaiTech-A dataset. To summarize, our proposed SRN+PS method performs better than most of the exiting methods on dense crowd datasets.

For sparse examples, in Tables 1 and 3, we perform evaluation on two datasets: ShanghaiTech Part_B and UCSD. Our proposed method outperforms most of the previous methods except for MCNN with respect to MAE on the UCSD dataset (our MAE is 0.17 higher). Apart from using and combining features between adjacent scales, the SRN+PS method focuses on object regions accurately by point supervision to segment spatial features. In addition, a novel loss function, the row vector-based counting loss, forces the model to optimize for the crowd count estimation. Thanks to these improvements, the SRN+SP method achieves better results on both datasets: the MAE/MSE on the ShanghaiTech Part_B and UCSD datasets are 13.8/18.8 and 1.24/1.63.

In the crowd analysis of weakly supervised learning, we follow the experiment methodology in the pioneering work from Sam et al. [9], which is one of the first attempts in this direction. We compare the performance of SRN+CWTA with that of other methods from Sam et al. [9] in Table 4. Although the performance of our SRN+CWTA is slightly worse (MAE is higher by 4) on Part_A dataset, SRN+CWTA is robust (MSE is lower by 6.1) and has significant advantages on more dense crowd datasets such as the UCF_CC_50. Its performance even exceeds the CCNN supervised method [9] in some cases. This is likely due to the fact that multi-scale structure can obtain more accurate salient object spatial information in unsupervised learning, especially in dense crowd estimation and analysis. This experiment shows that our SRN structure can be successfully applied in weakly supervised crowd analysis.

For the quality analysis of density maps, we get the PSNR and SSIM as defined in Eq. (11). The estimated results are shown in Table 5, which shows that the SRN+PS achieves the highest PSNR and SSIM on the ShanghaiTech Part_A dataset. The overall results generated by the SRN+PS method on all three datasets are given in Table 6.

To evaluate the practicability of crowd scene analysis methods in real-world applications, we analyze the complexity of each model in Table 7. The CP-CNN model has the most parameters, which is 500 times more than the least, the MCNN method, limiting its applications. In contrast, the SRN+PS has 13.2 million parameters, which is in the middle of the spectrum. Therefore, although our method performs slightly worse than the CP-CNN on the ShanghaiTech Part_A dataset, it gains significantly in terms of computational cost. Our SRN+PS model runs on an Intel Core i7 PC with a NVIDIA GTX1070 GPU and uses TensorFlow as backend.

4.4. Ablations on the shanghaitech parta dataset

Our proposed SRN structure benefited from two major improvements: the point supervision module and the row vector-based counting loss function. Therefore it is necessary to compare the performances of our method with and without these two parts, which are shown in Table 8. By using point supervision modules of three different scales in the SRN model, our performance on the ShanghaiTech Part_A dataset get significantly better, with the MAE/MSE 29.4/25.2 lower than that without point supervision modules. In addition, we weight the row vector-based counting

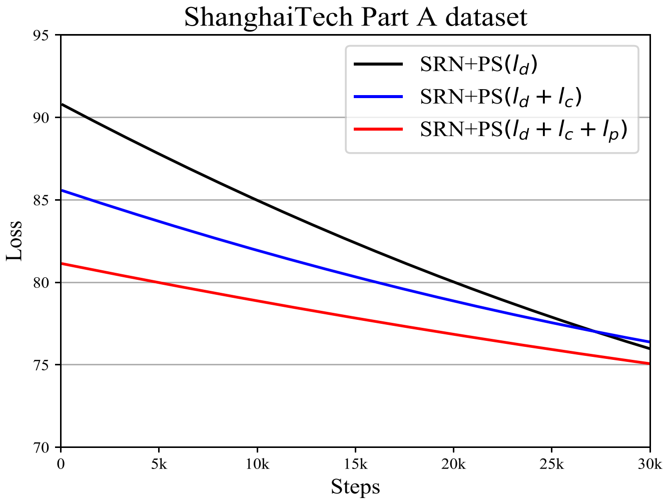


Fig. 6. The training loss trajectories of the SRN with the point supervision module, the row vector-based counting loss l_c, and the perceptive loss l_p.

loss l_c on euclidean loss l_d, and it achieves significant improvements with the MAE/MSE 26.9/17.9 lower than that without l_c. From the last row of Table 8, we can see that the perceptive loss l_p further optimizes the crowd counting results of the SRN with the above two assistant modules. These results imply that the above optimization modules can be extended to other crowd estimation networks to improve predictive performance.

In addition, we illustrate in Fig. 6 the differences in training losses of our method with point supervision module, row vector-based counting loss l_c, and perceptive l_p. In the early stage of training, the loss trajectory of SRN+PS with the counting loss is lower than that of the SRN+PS. However, their trajectories quickly converges to be similar as training progresses. This may be because the counting loss only focuses on the accuracy of crowd counting, but it increases the quality loss of density maps. The perceptive loss (the red curve in Fig. 6) can tackle this problem well due to high-level perceptual features of the predicted and ground-truth density maps at different scales of the SRN from a pre-trained VGG-16 model at layer of relu2_2. Our objective is to minimize the perceptual differences between the above two images. Therefore, the weighted sum of the three loss functions (l_d, l_c, and l_p) is the solution to optimize the training process. We also analyze the impact of the stack pooling module on the predicted results of SRN. In the last two rows of Table 8, the performances of SRN are further improved by using the stack pooling layer, with the MAE/MSE 4.6/14.6 lower than that with only the single pooling on the ShanghaiTech Part_A dataset. These results indicate that the stacked pooling layer is an effective module for crowd scene analysis tasks.

4.5. Visualization of results

To visualize the prediction of our SRN+PS model, we display the point maps and density maps generated from our methods in fully supervised and weakly supervised learning. Figs. 7 (from Part_A) and 8 (from Part_B and UCSD) show the sample results in dense and sparse crowd scenes respectively. In each figure, the first row shows the test images, and rows 2–4 are the density maps generated from the SRN+PS, the predicted point maps, and the ground-truth density maps. To illustrate the crowd count and quality of density maps clearly, we adds the counting, PSNR and SSIM values below the images in each column. As shown in both figures, our method achieves better accuracy in both dense and sparse crowd scenes, and the predicted point maps cover the head regions well. There are some cases where the results are not as good as expected, such as column 4 in Fig. 7 (SSIM is only 0.55) and column 2 in Fig. 8 (SSIM is only 0.55), both of which are extreme cases (extremely dense and extremely sparse). This is a problem for future research. We also give the predicted density maps of the weakly supervision framework based on Fig. 4. Fig. 9 (from Part_A) shows the rough density maps by SRN+CWTA. Although the density maps are far from those obtained by fully supervised learning

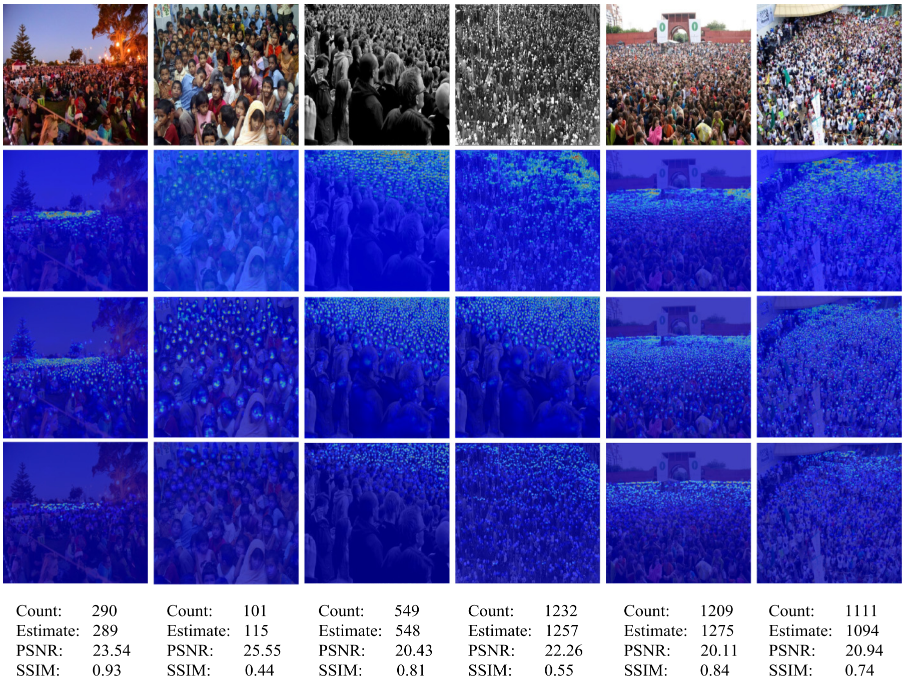


Fig. 7. Examples of predicted results in dense crowd scene.

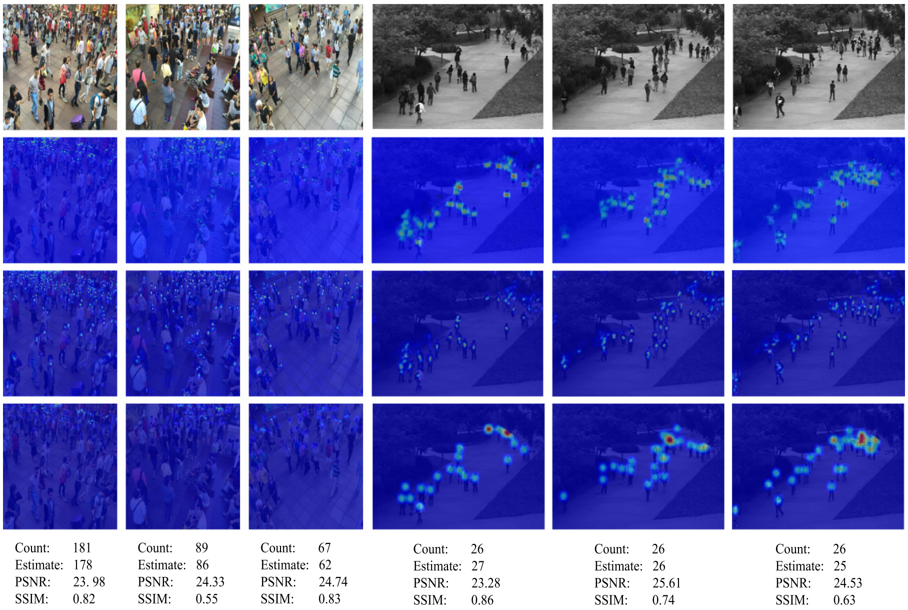


Fig. 8. Examples of predicted results in sparse crowd scenes.

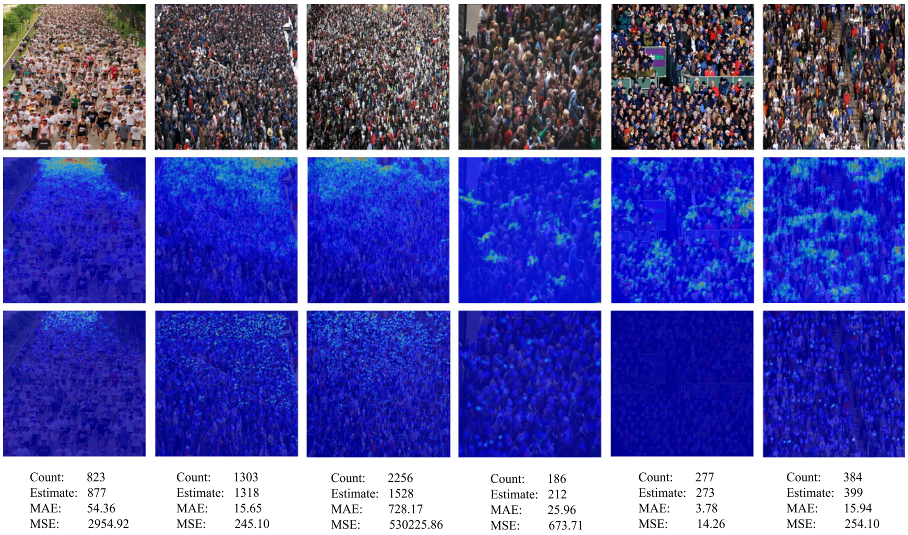
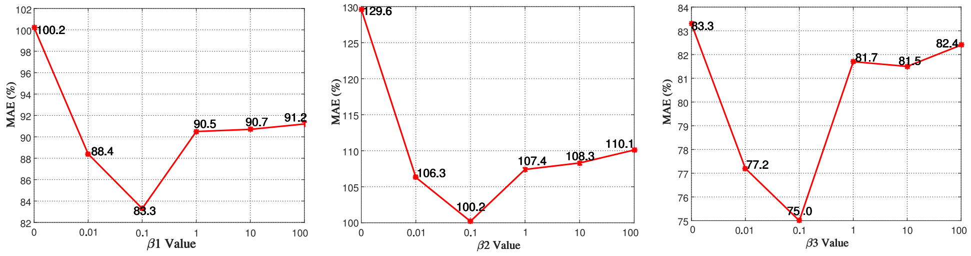


Fig. 9. Examples of weakly supervised learning predictions.



(a) (b) (c)

Fig. 10. Comparisons of MAE for different β_1, β_2, and β_3 values on the ShanghaiTech Part_A data.

methods in terms of quality, they are close to the results predicted by fully supervised learning in terms of crowd counting. One question worth future research in weakly supervised learning is how to use some modules such as attention mechanism to locate the head region accurately, in order to further bridge the gap between the two training paradigms in predicted density map quality.

4.6. Study of parameters β_1, β_2, and β_3

In order to choose the optimal values of β_1, β_2, and β_3 in Eq. (7), we perform comparative experiments on Part_A of the ShanghaiTech dataset. We firstly experiment on β_2 which controls the effect of the PS module on the performance of SRN, followed by an experiment on β_1 which is the weight of the row vector-base counting loss function based on SRN+PS. Finally, we study β_3 which is the weight of the perceptive loss function based on SRN+PS with l_c. As shown in Fig. 10, MAE errors in Fig. 10(a)–(c) decrease as the values of β_1, β_2, and β_3 increase, and the lowest error is obtained at 0.1. Therefore, we set β_1, β_2, and β_3 all to 0.1 in our experiments.

5. Conclusion

In this paper, we propose a novel model, called the Scale-Recursive Network with Point Supervision (SRN+PS), for crowd scene analysis, including head counting tasks and density map estimations. This model is a multi-scale architecture that utilizes the features of adjacent scales, which tackles the point segmentation task to boost the quality of density maps except counting estimation. In addition, the joint training strategy of multiple loss functions further improves the accuracy of counting predictions. Finally, a weakly supervised learning-based SRN+CWTA model is discussed to address the performance gap between fully supervised and unsupervised learning approaches. Experiments show that our method is robust in both dense and sparse crowd scene density estimations, and can be incorporated into other related networks.

As shown in our paper, the crowd scene analysis contains not only the crowd counting prediction, but also the location of objects and discrimination areas. Therefore, one promising direction for future research is to incorporate discriminative information to allow SRN to learn more powerful feature representations, an idea inspired by Cheng et al. [35] and Zhou et al. [36]. Besides, we plan to take advantage of local and multi-scale contextual information to predict crowd counting more precisely.

Declaration of Competing of Interest

The authors declared that they have no conflicts of interest to this work. We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

CRediT authorship contribution statement

Zihao Dong: Research concept and design, Collection and assembly of data, Data analysis and interpretation, Experiments, Writing the article, Critical revision of the article. **Ruixun Zhang:** Revision of the article. **Xiuli Shao:** Final approval of article. **Yumeng Li:** Revision of the article.

References

[1] D. Onoro-Rubio, R.J. López-Sastre, Towards perspective-free object counting with deep learning, in: Proceedings of the European Conference on Computer Vision, Springer, 2016, pp. 615–629.

[2] Y. Zhang, D. Zhou, S. Chen, S. Gao, Y. Ma, Single-image crowd counting via multi-column convolutional neural network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 589–597.

[3] L. Boominathan, S.S. Kruthiventi, R.V. Babu, Crowdnet: A deep convolutional network for dense crowd counting, in: Proceedings of the Twenty-forth ACM international conference on Multimedia, ACM, 2016, pp. 640–644.

[4] Z. Shen, Y. Xu, B. Ni, M. Wang, J. Hu, X. Yang, Crowd counting via adversarial cross-scale consistency pursuit, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 5245–5254.

[5] Y. Li, X. Zhang, D. Chen, Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 1091–1100.

[6] Y. Zhang, C. Zhou, F. Chang, A.C. Kot, Multi-resolution attention convolutional neural network for crowd counting, Neurocomputing 329 (2019) 144–152.

[7] J. Liu, C. Gao, D. Meng, A.G. Hauptmann, Decidenet: counting varying density crowds through attention guided detection and density estimation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 5197–5206.

[8] A. Makhzani, B.J. Frey, Winner-take-all autoencoders, in: Proceedings of the Advances in Neural Information Processing Systems, 2015, pp. 2791–2799.

[9] D.B. Sam, N.N. Sajjan, H. Maurya, R.V. Babu, Almost unsupervised learning for dense crowd counting, in: Proceedings of the Association for the Advancement of Artificial Intelligence, 2019.

[10] V.A. Sindagi, V.M. Patel, A survey of recent advances in CNN-based single image crowd counting and density estimation, Pattern Recogn. Lett. 107 (2018) 3–16.

[11] H. Idrees, M. Tayyab, K. Athrey, D. Zhang, S. Al-Maadeed, N. Rajpoot, M. Shah, Composition loss for counting, density map estimation and localization in dense crowds, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 532–546.

[12] D. Kang, Z. Ma, A.B. Chan, Beyond counting: comparisons of density maps for crowd analysis tasks-counting, detection, and tracking, IEEE Trans. Circ. Syst. Video Technol. (2018).

[13] R. Stewart, M. Andriluka, A.Y. Ng, End-to-end people detection in crowded scenes, in: Proceedings of the IEEE Conference on Computer Vision and Pattern recognition, 2016, pp. 2325–2333.

[14] E. Lu, W. Xie, A. Zisserman, Class-agnostic counting, arXiv preprint arXiv:1811.00472 (2018).

[15] K. Kang, X. Wang, Fully convolutional neural networks for crowd segmentation, arXiv preprint arXiv:1411.4464 (2014).

[16] I.H. Laradji, N. Rostamzadeh, P.O. Pinheiro, D. Vazquez, M. Schmidt, Where are the blobs: Counting by localization with point supervision, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 547–562.

[17] G. Cheng, P. Zhou, J. Han, Learning rotation-invariant convolutional neural networks for object detection in vhr optical remote sensing images, IEEE Trans. Geosci. Remote Sens. 54 (12) (2016) 7405–7415.

[18] G. Cheng, J. Han, P. Zhou, D. Xu, Learning rotation-invariant and fisher discriminative convolutional neural networks for object detection, IEEE Trans. Image Process. 28 (1) (2018) 265–278.

[19] S. Huang, X. Li, Z.-Q. Cheng, Z. Zhang, A. Hauptmann, Stacked pooling: improving crowd counting by boosting scale invariance, arXiv preprint arXiv:1808.07456 (2018).

[20] D.B. Sam, S. Surya, R.V. Babu, Switching convolutional neural network for crowd counting, in: Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2017, pp. 4031–4039.

[21] V.A. Sindagi, V.M. Patel, Generating high-quality crowd density maps using contextual pyramid CNNS, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 1861–1870.

[22] C. Zhang, H. Li, X. Wang, X. Yang, Cross-scene crowd counting via deep convolutional neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 833–841.

[23] L. Zhu, Z. Zhao, C. Lu, Y. Lin, Y. Peng, T. Yao, Dual path multi-scale fusion networks with attention for crowd counting, arXiv preprint arXiv:1902.01115 (2019).

[24] Z. Zhao, H. Li, R. Zhao, X. Wang, Crossing-line crowd counting with two-phase deep neural networks, in: Proceedings of the European Conference on Computer Vision, Springer, 2016, pp. 712–726.

[25] G.E. Hinton, R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 313 (5786) (2006) 504–507.

[26] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, B. Frey, Adversarial autoencoders, arXiv preprint arXiv:1511.05644 (2015).

[27] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, W.-c. Woo, Convolutional LSTM network: a machine learning approach for precipitation nowcasting, in: Proceedings of the Advances in Neural Information Processing Systems, 2015, pp. 802–810.

[28] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 2980–2988.

[29] J. Johnson, A. Alahi, L. Fei-Fei, Perceptual losses for real-time style transfer and super-resolution, in: Proceedings of the European Conference on Computer Vision, Springer, 2016, pp. 694–711.

[30] H. Idrees, I. Saleemi, C. Seibert, M. Shah, Multi-source multi-scale counting in extremely dense crowd images, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2013, pp. 2547–2554.

[31] A.B. Chan, Z.-S. J. Liang, N. Vasconcelos, Privacy preserving crowd monitoring: counting people without people models or tracking, in: 2008 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2008, pp. 1–7.

[32] V.A. Sindagi, V.M. Patel, Cnn-based cascaded multi-task learning of high-level prior and density estimation for crowd counting, in: Proceedings of the 2017 Fourteenth IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), IEEE, 2017, pp. 1–6.

[33] S. Zhang, G. Wu, J.P. Costeira, J.M. Moura, Understanding traffic density from large-scale web camera data, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5898–5907.

[34] S. Zhang, G. Wu, J.P. Costeira, J.M. Moura, Fcn-rlstm: deep spatio-temporal neural networks for vehicle counting in city cameras, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 3667–3676.

[35] G. Cheng, P. Zhou, J. Han, Duplex metric learning for image set classification, IEEE Trans. Image Process. 27 (1) (2017) 281–292.

[36] P. Zhou, J. Han, G. Cheng, B. Zhang, Learning compact and discriminative stacked autoencoder for hyperspectral image classification, IEEE Trans. Geosci. Remote Sens. (2019).

Zihao Dong received his M.S. degree in Control Science and Engineering from Tianjin University of Technology, China, in 2016. Now, he is currently pursuing his Ph.D. degree in Computer Science from Nankai University, China. His research interests include object sematic and instance segmentation, edge detection, computer vison and machine learning.

Ruixun Zhang received his Ph.D. degree in Applied Mathematics from MIT in 2015 and BS in Statistics and B.A. in Economics from Peking University in 2011. Currently he is a research affiliate at the MIT Laboratory for Financial Engineering. His research interests include data science and financial engineering.

Xiuli Shao received her Ph.D. degrees in Control theory and Control Engineering from Nankai University, China, in 2002. Now, she is a professor and Ph.D. tutor in the College of Computer Science, Nankai University. Her research interests are artificial intelligence, data analysis, and software engineering.

Yumeng Li received her B.S. degrees in computer science and technology from Beihua University, China, in 2014. Now, she is currently pursuing her master degree in computer science and technology at Nankai University. Her research interests include computer vision, object detection and machine learning.